

JVM Crash Guard

On this page:

- [Start Monitoring for JVM Crashes](#)
- [Watch the Video](#)

Related pages:

- [Standalone Machine Agent](#)

When a JVM crash occurs, you need to be notified as soon as possible. Knowing about a JVM crash is critical because it may be a sign of a severe runtime problem in an application. Furthermore, you may want to take remediation steps when a crash event has occurred.

JVM Crash is an event type implemented as part of JVM Crash Guard that can provide you with critical information for handling JVM crashes.

The following image shows the Events window with two JVM Crash events:

The screenshot shows the AppDynamics Events window for 'cart-load-1_JMS_NODE'. The left sidebar is set to 'Tiers & Nodes'. The main area displays a table of events with columns: Type, Summary, Time, Business Transacti..., Tier, No..., and Actions. Two 'JVM Crash' events are visible, both with a red warning icon. The first event occurred at 03/18/14 2:25:10 PM, and the second at 03/18/14 2:15:40 PM. Between them are four 'App Server Restart' events with blue information icons, each with a summary of 'Application Server JVMwas re-started No...'. The table also shows columns for 'Business Transacti...', 'Tier', and 'No...' with partial values like 'Order P...', 'Invento...', and 'EComm...'. A search bar and 'Show' button are at the top right of the table area.

Type	Summary	Time ↓	Business Transacti...	Tier	No...	Actions
JVM Crash	JVM Crash detected	03/18/14 2:25:10 PM	-	Order P...	Node_8...	⚡
App Server Restart	Application Server JVMwas re-started No...	03/18/14 2:20:58 PM	-	Invento...	Node_8...	-
App Server Restart	Application Server JVMwas re-started No...	03/18/14 2:20:58 PM	-	Order P...	Node_8...	-
App Server Restart	Application Server JVMwas re-started No...	03/18/14 2:20:57 PM	-	EComm...	Node_8...	-
App Server Restart	Application Server JVMwas re-started No...	03/18/14 2:20:57 PM	-	EComm...	Node_8...	-
JVM Crash	JVM Crash detected	03/18/14 2:15:40 PM	-	Order P...	Node_8...	⚡

Double-click the JVM Crash event on the Events window to see more information on troubleshooting the cause of the JVM crash.

To perform root-cause analysis and troubleshooting:

- In the Events window, double-click the JVM Crash of interest.
- Examine any logs associated with the JVM Crash event. (The local log folder is determined by the type of JVM and how it is configured. For more information, refer to the documentation for the specific JVM.)

The JVM Crash window also displays information about actions executed as a result of the crash. These are actions that you specify when creating the policy that is triggered by a JVM crash event.

JVM Crash

Summary Details Actions Executed (0) Comments (0)

Copy to Clipboard

Command line
 /Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home/jre/bin/java -Ddoes.not=matter1 -Ddoes.not=matter2 -javaagent:/Users/bwinslow/git/cart-imp/TIER1TOMCAT/appagent/javaagent.jar -Dappdynamics.agent.uniqueHostId=cart-machine -Dappdynamics.bciengine.write2disk=/Users/bwinslow/git/cart/bci -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/Users/bwinslow/git/cart-imp/TIER1TOMCAT/D-XX:+UseG1GC -XX:PermSize=256m -XX:ErrorFile=/Users/bwinslow/crashlogs/java_error%p.log -Xmx512m -Dminimum_age_for_evaluation_in_mins=0 -DlistenPort=8000 -Dshutdownport=8180 -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.util.logging.config.file=/Users/bwinslow/git/cart-imp/TIER1TOMCAT/conf/logging.properties -Dcom.appdynamics.bounded.collections.default.policy=WarmIfBoundExceeded -Dcatalina.base=/Users/bwinslow/git/cart-imp/TIER1TOMCAT -Dcatalina.home=/Users/bwinslow/git/cart-imp/TIER1TOMCAT -Dappdynamics.viewer.port=8990 -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9004 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Dappdynamics.enable.missed.class.scan=false -Dappdynamics.enable.missed.class.scan.freq=120 -Dappdynamics.avoid.boot.class.loader.lookup=true -Dappdynamics.background.transaction.reporter.queue.type=circular -Dappdynamics.async.all.metrics=sometimes -Dappdynamics.background.transaction.reporter.initial.circular.queue.size=1000 -Dappdynamics.hotspot.enabled=false -Dappdynamics.bciengine.should.implement.new.interfaces=true -Dappdynamics.socket.collection.bci.enable=false -Dappdynamics.agent.runtime.dir=/Users/bwinslow/AgentRuntime -Dappdynamics.object.monitor.hang.detect=600 -Dappdynamics.scheduler.time.adjuster.required=sync -classpath /Library/Java/JavaVirtualMachines/jdk1.7.0_45.jdk/Contents/Home/lib/tools.jar:/Users/bwinslow/git/cart-imp/TIER1TOMCAT/bin/bootstrap.jar org.apache.catalina.startup.Bootstrap

Crash Date/Time
 Thu Mar 27 15:55:43 PDT 2014

Crash Reason
 SIGSEGV (0xb) at pc=0x00000010e5046fd, pid=36832, tid=67075

File_0
 1395960944535/java_error36832.log

Host name
 osx1lbwins.local

IP Address
 192.168.1.9

PID
 36832

appName
 ACME Book Store Application

nodeName
 Node_8000

tierName
 ECommerce Server

Close

The JVM Crash event captures the following information: timestamp, crash reason, host name, IP address, process ID, application name, node name, and tier name and displays them on the details page.

In the Crash Reason details field of the JVM Crash Details tab, the JVM Crash details indicate the root cause of the crash if available; for example, a java.lang.OutOfMemoryError or Segmentation Fault.

To facilitate the discovery and display of the reason for the JVM crash, JVM Crash Guard provides full support for:

- Hotspot JVM error log analysis
- IBM JVM System Dump log analysis
- Jrockit JVM error log analysis

Start Monitoring for JVM Crashes

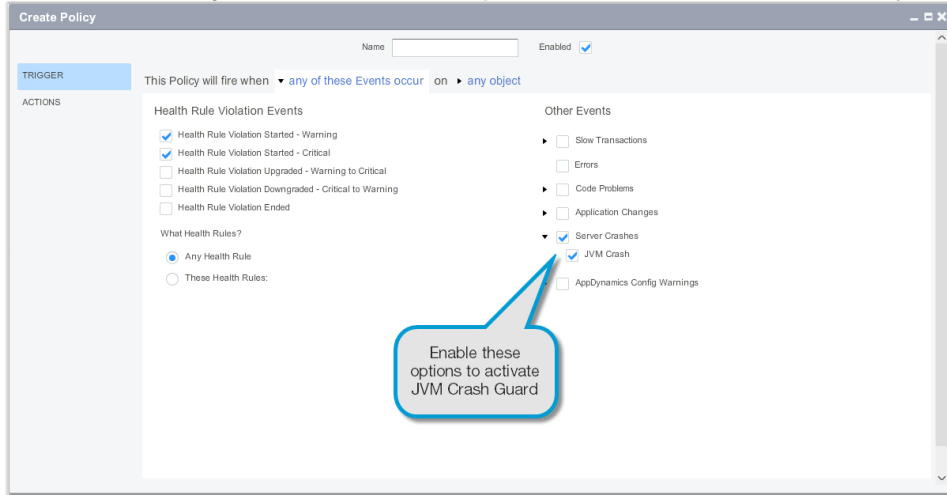
JVM Crash Guard works with the [Standalone Machine Agent](#) to trigger an AppDynamics policy when a JVM Crash event occurs. You must therefore have a Machine Agent installed and enabled on the machine that you want to monitor for JVM crashes.

Additionally:

- On Windows, the Machine Agent must run in Administrator root mode.
- On Linux, JVM Crash Guard requires that the Machine Agent user be able to read all the processes in `/proc/*`. This may be the 'root' user or another user with this privilege.

Once you have verified the requirements, follow these steps to create a policy for JVM crash events:

1. From the left-hand navigation menu, click **Alert & Respond > Policies** and then click **Create a Policy**.



2. In the **Other Events** section, expand the **Server Crashes** option and click **JVM Crash**.
The JVM Crash event then becomes a trigger to fire a policy.
3. Proceed as usual to create the Policy. For more information on creating Policies, see [Policies](#).

Note: If an uninstrumented JVM crash happens within less than a minute of a previous crash then it will not be reported by the Standalone Machine Agent. In some circumstances, the JVM may crash and then be restarted only to crash again within one minute. For this repetitive cycle crash and restart scenario, only the first JVM crash is reported by the agent.

Watch the Video

For full-screen viewing, click [Standalone Machine Agent - Overview of JVM Crash Guard](#).